Generation of variability in shape, aspect and time of 3D Fruits and Vegetables

Adrien Verhulst, Jean-Marie Normand, Guillaume Moreau

Generation of variability in shape, aspect and time of 3D Fruits and Vegetables

Adrien Verhulst*, Jean-Marie Normand*†, Guillaume Moreau*†
*AAU UMR CNRS 1563,
École Centrale de Nantes, France.
firstname.lastname@ec-nantes.fr
†Inria Hybrid, France.

***Abstract*— With the growth of organic Fruits and Vegetables (FaVs) markets, there is now a trend in marketing research toward studies of *non-standardized fruits and vegetables* in stores. Yet, because of the decaying nature of FaVs, it is difficult to conduct such studies. A solution is to conduct them within a Virtual Environment (VE) (with virtual FaVs). Therefore, it is of interest to develop an approach to generate a large variety and variability of FaVs, so one can later use them in a VE. In this paper we introduce a pipeline to generate a large variability of FaVs, focusing both on their shape and on their external appearance. Regarding the shape, we use a semi-automated approach. A parametric Skeletal Structure and Generalized Cylinders (GCs) generates their overall shape and metaball-based techniques give them an organic aspect. Regarding their external appearance, we use a particle system approach to simulate their modifications over time. This particle system-based approach decomposes FaVs appearance changes into distinct visual characteristics producing different texture maps that can be combined.**

***Index Terms*—Computer Graphics, Appearance Variability, Shape Variability, Maturing, Aging Process**

I. Introduction

According to the Food and Agriculture Organization of the United Nation (FAO) [11], about 45% of Fruits and Vegetables (FaVs) are "food losses" in Europe; mostly due to the post-harvest FaVs classification set by quality standards (such as the Organization for Economic Cooperation and Development (OECD) FaVs standards) [26].

To reduce the "food losses", a recent trend consists of selling abnormal (i.e., non-standards) FaVs by emphasizing on organic or pro-environmental considerations [21]. Yet, marketing studies aiming to study the impact of abnormal FaVs on consumer behaviors in retail settings are difficult to perform since (i) such FaVs are not always available, (ii) their appearance is time dependent. Fortunately, several studies have shown that marketing studies were possible with the help of Virtual Reality (VR) techniques [31], [33], [34].

However in order to perform those studies in VR, it is necessary to have a large range of FaVs 3D models showing both shape and aspect variabilities in a plausible way.

While it is fairly easy to 3D model individual FaVs, it is time-consuming to manually produce a large variability of FaVs (i.e., with different sizes, shapes). Even if well known techniques such as the *cage deform* [19] can increase the variation of a base object, those techniques cannot handle all the possible variations of the FaVs (e.g., growths cannot be modeled since they "add matter" to the base object). To generate an important number of different FaVs, a previous work was based on a custom *3D L-system* [6]. We follow a somehow similar approach by using a *skeletal structure*, generalized cylinders (GCs) and *metaballs*. The skeletal structure allows for a large range of shapes; the generalized cylinders then sweep along the skeletal structure to give the overall shape; finally metaballs are used to obtain an organic-looking shape.

Moreover, throughout their life-cycle FaVs experience several appearance changes (maturing, aging, diseases, etc.) depending on their genotypes and on their environmental factors (such as chemical, physical and biological phenomena [23]). As a consequence, FaVs' external appearance (i.e., color, visual characteristics) is time-varying, making it difficult and time-consuming to do it manually. Even if several texture-based computer graphics techniques can alter the visual appearance of 3D objects, to the best of our knowledge those techniques are not suitable to render FaVs' complex changes of appearance. Therefore, to make those, our approach uses *particle-systems* and allows the user to separate a "complex" appearance change into several "simple" visual characteristics each of which is managed by a particle system (in the rest of this paper we use the term *p-system* for particle system).

In this paper, our goal is (i) to present a customizable approach to generate visually-realistic FaVs with variability in shape and aspect; (ii) to generate FaVs with malformations and a "non-perfect" aspect; (iii) to generate FaVs appearance changes (i.e., at different maturation stages) in a visually realistic way. Ultimately, we want to use this approach to populate a virtual market with abnormal FaVs for marketing studies purposes. It is important to precise that we do not try to generate FaVs in real-time nor to obtain a 3D mesh with a limited number of vertices or a good edge flow.

In the remainder of this paper, we first review existing work on FaVs generation and FaVs appearance changes. We introduce our approach, i.e. 3D mesh generation in Section III-A and 2D texture generation in Section III-B. Results are presented in Section IV before drawing conclusions and presenting future work in Section V.

II. Related Work

We present a literature review regarding FaV 3D mesh generation, FaV time-varying appearance of natural phenomena

and FaV deformations in computer graphics.

Since our approach uses (1) a skeletal curves associated with GCs (typically, a sweep surface) to obtain a "base shape" and then (2) implicit surface (in our case, metaballs) to merge overlapping GCs along the base shape, we first begin by a short review of skeletal structures, Generalized Cylinders (GCs) and metaballs.

A. Skeletal structures, generalized cylinders and metaballs

Skeletal structures can be use to procedurally model complex objects from a simpler representation, and as such have been widely used to produce organic shapes.

Bloomenthal et al. [5] already used skeleton curves (along GCs and 3D surfaces algebraic representation techniques, inspired from Blinn [4]) to model botanical trees. More recent techniques have been proposed to generate base meshes of 3D articulated shapes from a skeletal structure. Ji et al. [18] proposed the so-called *B-meshes*. Their method swaps "shape balls" along a skeleton and approximates a mesh using the swapped shape balls' convex hull. The resulting mesh has improved tessellation over traditional metaballs modeling methods.

Following [18], Bærentzen et al. [1] proposed the *Skeleton to Quad-dominant polygonal Mesh algorithm* (SQM). Their method generates polyhedra at each branch node, which are refined and connected with tubes consisting of a single loop of quadrilaterals. The resulting mesh has an improved edge flow compared to [18]. B-meshes and SQM methods can arguably be used in our case (cylindrical shapes are ubiquitous in living nature as noted by the biologist Wainwright in 1988 [32]), but both methods can only produce cylindrical shapes while our method is more flexible. Since several common FaVs (e.g., beefsteak tomato, bell pepper, etc.) cannot be easily modeled by combining cylinders only, we prefer to use GCs.

While we found very little work using GCs to model FaVs, they were previously used to model plants, Bloomenthal et al. [5] used GCs to model botanical tree branches. More recently, Prusinkiewicz et al. [29] used GCs along with a software called L-studio[1] to model complex plants in a "global-to-local" fashion. L-studio is a plant simulation and visualization software using L-systems. An L-system (or Lindenmayer system) is a type of formal grammar and a rewriting system used to represent plants growth or development. It generally consists of an alphabet of symbols and a set of production rules that are used to produce strings of symbols. The strings of symbols can then be converted into geometrical shapes and displayed. Fuhrer et al. [13] also used GCs with L-systems in order to generate hairy plants to improve the overall realism.

Metaballs [4], [25] have been extensively used to represent organic shapes and natural phenomena (mainly fluids), since the metaballs allow for a very "organic-looking" blending. A review by Pan et al. [27] presents some of metaballs' applications in the modeling of the human body, but more specific organic objects can also be represented (e.g., tumors [2] or organs [28]). To the best of our knowledge, there have been no attempt to model FaVs from metaballs.

[1]http://algorithmicbotany.org/lstudio/

B. Generation of FaVs' 3D models

The problem of generating automatically 3D models of FaVs has been previously tackled in various ways. Several studies were based on models of specific biological phenomena occurring in FaVs. In a pioneering work, Fowler et al. [12] presented an approach to model in 3D what is called "spiral phyllotaxis" (i.e., the arrangement of leaves on a plant stem) using a collision-based model to position the organs of the fruit. More recently, Génard et al. [14], [15] modeled the behavior of a fruit, to later explore the biological processes impacting its quality. Following a similar goal, Cieslak et al. [8], [9] generated a 3D mesh of a fruit using a magnetic resonance imaging scanner.

Others studies modeled 3D FaVs with fewer concerns about their biological accuracy. Huang et al. [17] used custom L-systems to procedurally generate realistically-looking grape bunch. Bohl et al. [6] recently proposed an approach using a 3GMap L-system: a grammar with a specific set of rules constructing a 3D mesh and then subdividing it. They tested their approach on several FaVs and represented different species of tomatoes with a large range of variability. One of the advantages of their method is the possibility to model not only the outside shape of the FaV but also its inner parts. Although this work provides so far the best results regarding 3D modeling of FaVs' malformations, the complexity and lack of interactivity of 3GMap L-system grammars makes it difficult to produce an important range of deformed FaVs, which is our main goal (indeed, each new deformation would require a new set of rules). Yet, our 3D mesh generation approach is inspired from their set of specific rules.

C. Time-varying appearance of FaVs

While time-varying phenomena can be used on various objects (buildings, cars, etc.), we focus here on work that was carried out on FaVs. Time-varying techniques applied to FaVs can be traced back to Blinn [3], where bump mapping was used to represent wrinkles of a strawberry. Yet, to the best of our knowledge since then there has been little work carried on FaVs' appearance changes over time. Indeed, simulations of aging phenomena mostly focus on realistic natural phenomena applied to others objects (mostly non-organic).

More recently, Gu et al. [16] proposed a data driven approach (through a *Space-Time Appearance Factorization model*) to decay various FaVs (tested on apples, bananas and oranges). This approach requires to obtain data samples of a specific appearance change (under controlled laboratory conditions) and must be repeated for each phenomenon one wants to reproduce. Kider et al. [20] proposed a particle-based technique (taking into account physical, chemical and biological processes through a reaction-diffusion model) to simulate FaVs (tested on apples, oranges and tomatoes). This approach is physically realistic but can only be used for specific decay processes for which the model is known and

lacks control in its execution. In Fan et al. [10], a rule-based technique using a reaction-diffusion model is proposed to simulate the "ring rot" disease (tested on apples, pears and jujubes). This approach is physically realistic but only simulates this very specific (although common) disease.

While not targeting physically-realistic objects, we aim at rendering more than one appearance change at a time.

III. Pipeline

As a reminder, our process follows two main steps: (i) the creation of a large variety of 3D FaVs meshes (cf. Subsection III-A) and (ii) the generation of 2D textures to visually simulate FaVs appearance changes (cf. Subsection III-B). Those steps are fully independent of each other, and increase the range of variability of FaVs generated by our method.

A. 3D Mesh Creation

Our method especially focuses on the generation of malformations (such as growths, bumps, etc.) that can be applied to FaVs. We believe that those malformations can be represented by relying on skeletal structures. Indeed they present two main advantages: (i) it is simpler to generate a FaV's skeletal structure than a FaV's 3D mesh; (ii) malformations can be easily represented by extending a base FaV's skeletal structure. The overall approach is presented in Fig. 1.

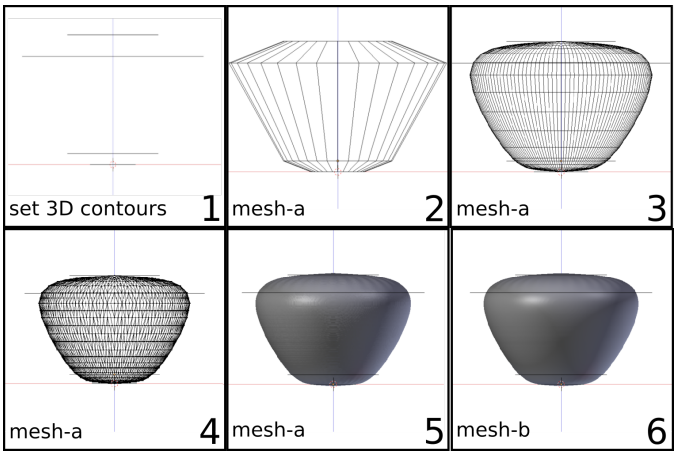

Fig. 1. From left to right and up to bottom: (1) place the cross-sections along the skeleton's nodes; (2) sweep the cross-sections along the skeletal structure, generate a 3D mesh noted `mesh-a`; (3) apply a Catmull-Clark subdivision [optional]; (4) triangulate `mesh-a` and subdivide edges above an arbitrary length; (5) generate and position a metaball at each vertex of `mesh-a`; (6) convert the metaballs in a 3D mesh noted `mesh-b` and smooth `mesh-b`.

Once the skeletal structure of a "base" FaV is defined (cf. Sec. III-A1), we construct the overall shape of the FaV using a GC-like technique. We define at each node of the skeletal structure a cross-section (without internal details), then each cross-section is swept along the skeletal structure (and eventually gradually morphs from the current cross-section to the next cross-section). At the end of our procedure, a 3D mesh of the FaV is generated (see Fig. 2).

If necessary, a *Catmull-Clark subdivision* [7] is applied to obtain a more "rounded" aspect. The `mesh-a` is triangulated

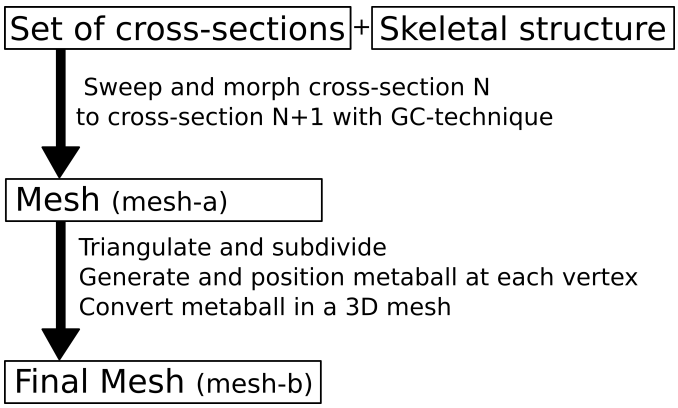

Fig. 2. Pipeline of the 3D mesh generation. From a skeletal structure, we generate a FaV 3D mesh.

and all its edges longer than an arbitrary threshold d are subdivided. A metaball is positioned at the position of every vertex of mesh-a. The radius of those metaballs must be greater than $\frac{d}{2}$. Using a marching cubes algorithm [22], the metaballs are converted into a 3D mesh and merged together into the final 3D mesh (called `mesh-b`). Since small imperfections are tolerable, we do not try to refine the placement of the metaballs nor their properties; yet a smoothing algorithm (a running average on each vertex position) is applied to `mesh-b`. Since we aim at using the FaV in an interactive VE, we eventually decimate it, i.e. reduce its number of vertices to optimize rendering time (we do not detail this step but this can be easily achieved by using an off-the-shelf 3D modeler such as Blender).

1) Input Generation details: The process (cf. Fig. 1) takes as input a skeletal structure (i.e, a *rooted tree*) and a set of cross-sections. Each cross-section is positioned at a corresponding node. There are as many cross-sections as vertices in the skeleton. A set of attributes (represented as a set of variations intervals) is stored for each node:

- The intensity and range that can be applied to the node's position and orientation (therefore modifying the curve in the skeletal structure).
- The intensity and range of a modification in its corresponding cross-section (e.g. scaling the cross-section).
- How likely a node ($node$) can instantiate a new node ($node_{new}$) (allowing us to randomly generate additional branches in the skeleton, which is helpful to model non-standards FaVs such as forked carrots).
- A spawning radius and a spawning weight for each attributes. If a node instantiates a new node $node_{new}$, then $node_{new}$ will be instantiated in the user-defined spawning radius of the current node? The new node $node_{new}$ is instantiated as a child of $node$ and inherits all of $node$'s attributes as well as $node$'s cross-section, all scaled by the user-defined spawning weight.

Variations are randomly applied within the user-defined intervals in order to easily produce a large range of FaVs

from a "base" FaV. Then, the cross-sections are swept along the skeletal structure. An example of skeletal structure (nodes and cross-sections) is presented in Fig. 3 and a result of our process applied to a Beefsteak tomato is illustrated in Fig. 4.

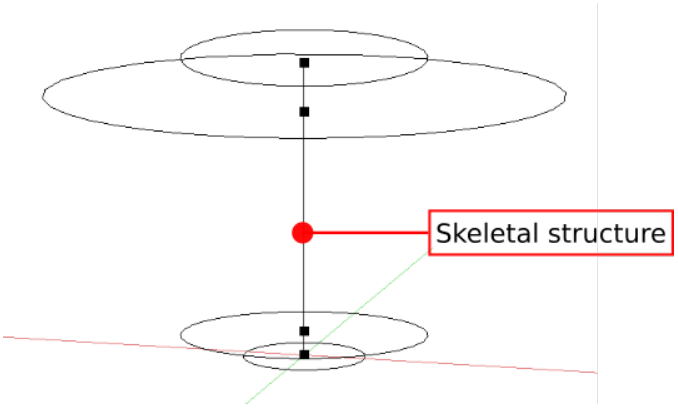

Fig. 3. A skeletal structure of a "base" apple. The center line is the skeletal structure. Each of its 4 nodes are displayed as well as their corresponding cross-sections.

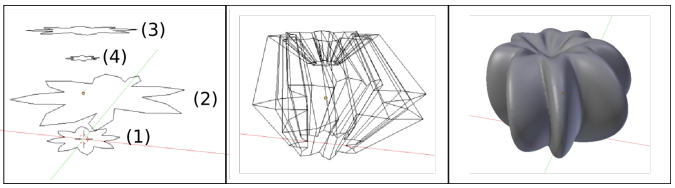

Fig. 4. Generation of a Beefsteak tomato.

B. 2D Texture Generation

Our 2D texture creation step takes into account physical, chemical and biological processes. Its execution (it is not, strictly speaking, a simulation) generates visually-realistic natural phenomena in FaVs (cf. Fig. 14). We believe that the majority of appearance changes (maturation, aging, diseases) of FaVs can be expressed as a composition of simple visual characteristics. Those can be categorized into overall shape, positioning and density (see Fig. 5).

Fig. 5. Visual characteristics of the ripening of a banana. We chose by observation three visual characteristics: (1) black dots, (2) black stripes (only in specific regions) and (3) black regions.

In order to visually represent a FaV appearance change, the user arbitrarily decomposes it into distinct visual characteristics (cf. Fig. 5). Each visual characteristic v is managed by a *p-system* which is in charge of generating a texture representing v. For each *p-system*, the user provides (1) the time of appearance of the phenomenon (with respect to the start of the algorithm), (2) the pace of emission and propagation of the particles, (3) the density of emission and propagation of the particles, (4) the regions of emission and propagation of the particles, and (5) the duration of the phenomenon.

Indeed, each *p-system* has three different behaviors: emission, propagation and maturation (cf. Section III-B1). Such behaviors, combined with the use of a texture representing the visual characteristic allow for a fine control of the overall natural phenomenon we aim at reproducing. Since each visual characteristic is controlled by a *p-system*, in order to produce the appearance change (i.e., the natural phenomenon represented as the composition of visual characteristics) our method relies on a *p-system manager*. The role of this *p-system manager* is to perform the simulation by (i) updating each of the *p-systems* at each step of the simulation and (ii) blending all the textures (produced by each *p-system*) together, cf. Fig. 6.

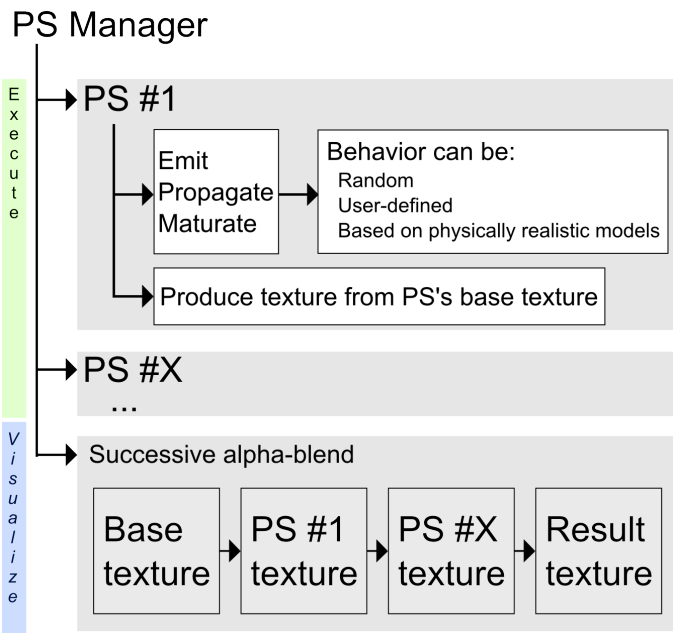

Fig. 6. Illustration of our framework (PS stands for *p-system*). The *p-system manager* goes through the *p-systems*, executes them and blends the textures that were produced(see examples in Fig. 13). Each *p-system* has an emission, a propagation and a maturation behaviors.

At each simulation step, a *p-system* has to perform the following actions (cf. Fig 7):

1) Determine where new particles can be emitted in the texture and where each particle can propagate itself on the p-system texture's pixels.
2) Calculate how many new particles n can be emitted and on how many pixels m each particle can propagate on.

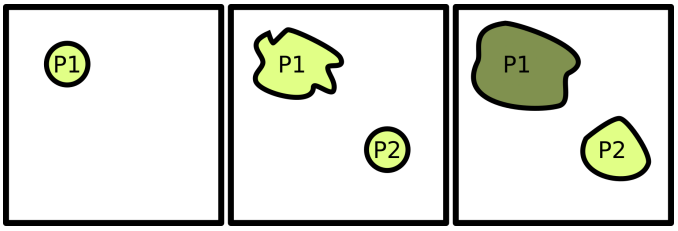

Fig. 7. From left to right: (1) a particle P1; (2) the particle P1 emits P2 and then propagates on the p-system texture's pixels; (3) both P1 and P2 propagate, then P1 matures and its pixels change color.

3) Emit, on the p-system texture, n new particles from already existing particles.
4) Propagate each particle on the p-system texture. Each particle propagate by covering near-by pixels.
5) Maturate each p-system texture's pixels covered by a particle (the pixel's color depends of the particle's p-system maturation model as well as the time since the pixel has been covered).

Note that the maturation step can change the behaviors of a particle, depending of the particle's life-time (i.e. the particle pass from a given *p-system* to another). Once each *p-system* has been updated, the manager blends the base texture of the FaV 3D model with the *p-systems*' textures (cf. Fig. 6). The alpha value depends of the particle's maturation value at a given pixel.

1) Implementation details: The p-system manager proposes a starting and ending simulation time, as well as a time-step value. Each p-system takes into parameter a given emission, propagation and maturation behavior, as well as a (1) an emission and a propagation mask, (2) a starting and an ending time. By default, both emission and propagation take into parameter (1) an emission (resp. propagation) rate ; (2) an emission (resp. propagation) distance ; (3) a probability map.

As mentioned above, by default, the maturation process can move a particle from one p-system to another (and therefore change the particle's behaviors). It is possible to extend those behaviors (by adding parameters and by implementing algorithms such as the one presented in [10]) so that appearance changes may be not only visually-realistic, but also physically-realistic.

IV. Results

In this section, we first give some details of implementation (see Section IV-A) before presenting results of the generation step (cf. Section IV-B) before illustrating our texture generation process (see Section IV-C).

A. Implementation details

We implemented our approach in Blender (v2.71) and Unity3D (v5.0), in python and C#, respectively for the 3D mesh creation and the 2D texture generation. The tests were performed on a simple laptop (CPU: Intel Core I5@2.4GHz processor; 8GB of DDR3 RAM; GPU: NVIDIA GeForce 840M). Regarding the 3D mesh creation, the computation time mainly depends of the metaballs' size (since the mesh's edges need to be subdivided accordingly). Regarding the 2D texture generation, the computation time at each step mainly depends on the complexity of the p-systems' behaviors as well as on the number of particles. Results were exported and rendered in Blender to take advantage of Blender's rendering engine.

B. Variability in shape of 3D FaVs

As mentioned previously, the goal of the generation step is to be able to produce in a semi-automated way misshapen FaVs. To the best of our knowledge, there exists no standards to classify the different malformations of FaVs (for example, the OECD standardizes FaVs [26] according to their commercial quality, but without consideration on the types of deformation). Therefore, when generating our FaVs we take inspiration from existing misshapen FaVs. We present below some of our results.

1) Apple malformation: The National Fruit Collection [24] characterizes apples along several parameters, such as size (small, medium, etc.) and shape (globular, flat, conical, etc.). Unfortunately, as mentioned earlier, there is no information regarding the classification of malformations in apples. We therefore present below a commercialized apple, the Granny Smith, characterized by a large size and "globosely conical" aspect (i.e, a globular form with a conical bottom).

In order to illustrate our approach, we first present skeletons of a "normal" (see Fig. 8, left) and of a "misshapen" Granny Smith (Fig.8, right). Adding a malformation is fairly easy, indeed, we only have to modify the skeletal structure of the "normal" Granny Smith. Final results with those skeletal structures are presented in Fig. 9.

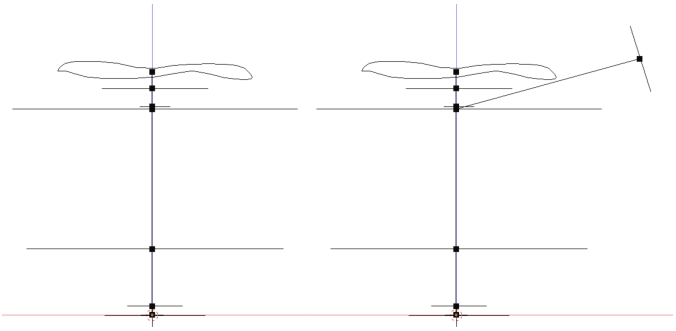

Fig. 8. Skeletal structures of a Granny Smith apple; Left: without malformation; Right: with a medium-sized growth (a variation of the left one).

2) Beefsteak tomato: More complex shapes can also be achieved using our process. We present as examples two Beefsteak tomatoes: a "normal" one and a misshapen one. The latter is inspired by a real photography we tried to reproduce (see Fig. 10). Our two automatically generated tomatoes are presented in Fig. 11).

3) Multiple FaVs: Finally, we present a set of different FaVs (apples, tomatoes, oranges, pears) automatically generated by our approach. Fig. 12 illustrates different fruits and vegetables: tomatoes, apples of two cultivar (Granny Smith

Fig. 9. Renders obtained from the skeletal structures presented in Fig. 8. Left: Granny Smith apple without malformation; Right: with a medium-sized growth.

Fig. 10. A real Beefsteak tomato ([30]).

and Fuji), oranges and pears (from top to bottom and left to right).

C. Variability in aspect and time of 3D FaVs

The previous section illustrated of our method can generated FaVs in a wide variety of shapes. Since the objective of our approach is to generate visually-realistic FaVs with variability both in shape and aspect, we present below results of our 2D texture generation approach. One of the advantages of our method is that the *p-systems* can be used to simulate many

Fig. 11. Two examples of misshapen Beefsteak tomatoes inspired from real images.

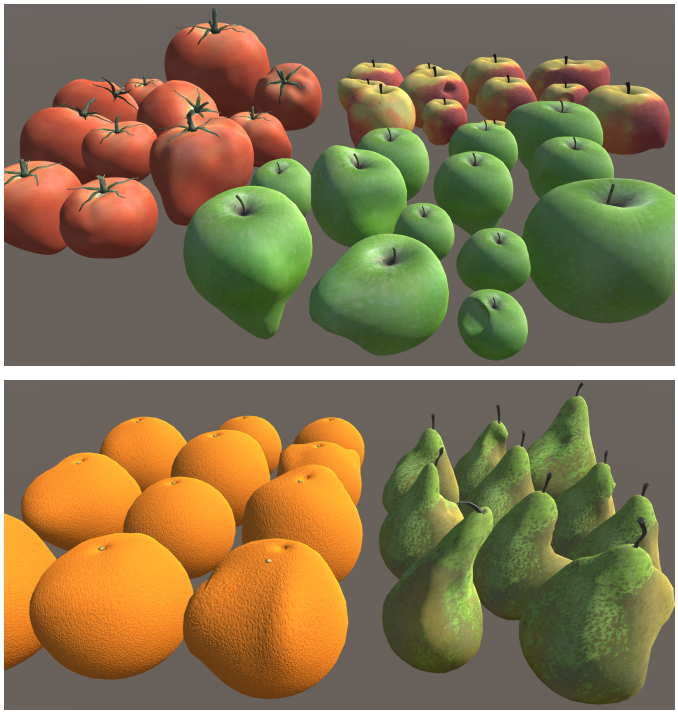

Fig. 12. Automatically generated FaVs, decimated in low-polygon models and rendered in a VR-ready 3D engine (Unity3D). Tomatoes, Apples, Oranges and Pears.

different phenomena such as aging (see Section IV-C1) but also diseases (see Section IV-C2).

1) Aging phenomenon: To illustrate our approach, we now detail the ripening of a banana. Based on the observation of this phenomenon in reality (see Fig. 5) we chose to decompose it into three visual characteristics: “dot”, “stripe” and “decay”. As a consequence we use three *p-systems*, cf. Fig. 13, to generate visually-realistic renderings of the banana.

The “dot”, “stripe” and “decay” p-systems are executed and generate at each simulation step three new textures (one for each p-system) representing the visual evolution over time of the banana ripening. In order to visualize this evolution, the p-system manager blends together those three textures and applies the result on a 3D banana model, cf. Fig. 14. The evolution of each p-system consists of consecutively: (1) emitting new particles, (2) propagating and (3) maturing particles. The maturation process can turn a particle from a p-system into another (e.g., a “dot” particle can evolve into a “decay” particle over time).

2) Diseases: As mentioned above, it is possible to simulate diseases by modifying the behaviors of the *p-systems*, we illustrate in this section results of two different diseases applied to our automatically generated FaVs. Our results are inspired by real world images.

Fig. 15 shows two non-standardized apples (according to the OECD) suffering from the Lenticel blotch pit disease (they could nevertheless be sold in organic markets). The reader is invited to compare our results to real photographies of apples

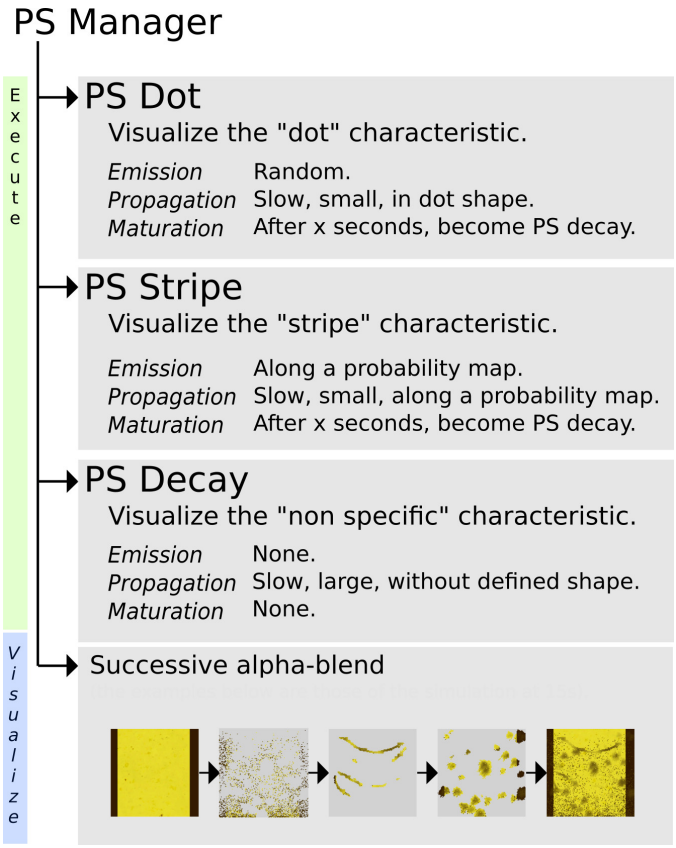

Fig. 13. Application of the 2D texture generation step to a banana's ripening. PS stands for p-system. Next to each behavior's name we briefly detail their implementation.

Fig. 14. Simulation result of a ripening banana.

suffering from such a disease[2]. In this case, there were two *p-systems* (P1 and P2). P1 emits new particles and propagates them. Particles mature and transform from P1 to P2. P2 does not emit new particles, nor propagate them, but change their color (from light brown to dark brown) by maturing them.

Fig. 16 shows a Beefsteak tomato suffering from Anthracnose disease (caused by a fungus). Again our implementation is inspired by real photographies[3].

In this case, there were two *p-systems*: the first one spreads dark particles upon the tomatoes, and, after a given time, some

[2]http://ephytia.inra.fr/fr/D/7437

[3]http://www.clemson.edu/extension/hgic/pests/plant_pests/veg_fruit/hgic2217.html

Fig. 15. Our Granny Smiths (cf. Fig. 9), suffering from Lenticel blotch pit. The resulting textures were also used as displacement map.

of these particles become part of the second p-system and turn yellow/orange. The resulting textures were also used as displacement maps.

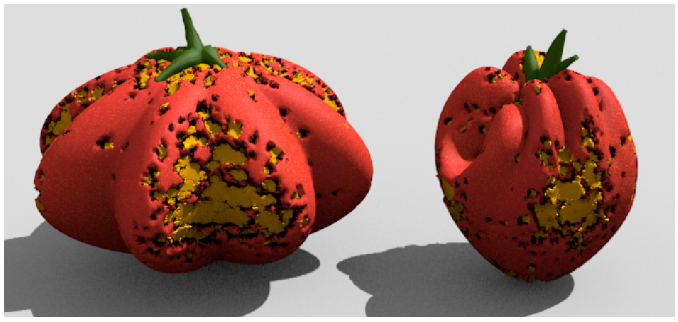

Fig. 16. Our Beefsteak tomatoes (cf. Fig. 11), suffering from Anthracnose.

V. Conclusion and Future Work

In this article, we presented an approach to generate visually-realistic FaVs with a large range of variability. To illustrate the pertinence of our approach we tested it on different FaVs. The skeletal structure allows for an easy control of FaVs' shape and of their shape variability. By modifying the skeletal structure, the user can easily generate FaV deformation (such as growths). Moreover, by separating appearance change into distinct visual characteristics, we allow the user to setup several p-systems (one for each visual characteristic) implementing emission, propagation and maturation behaviors. Each p-system gives rise to a texture representing a visual characteristic at given time of the simulation.

While it is not a simulation, our technique can be used to represent many different deformation and phenomena of a large variety of FaVs. In addition to FaVs visualization, our technique could also be used in different ways, e.g. following [20] resulting 2D textures could be used as data map, for example to map springs' stiffness of the mass-spring system in order to deform the FaV's 3D model, or, as we did in our renders, to provide data to our shader.

Our method still presents some drawbacks: not all FaVs can be generated using our 3D mesh generation step (e.g., lettuce

or grapes) and their number of vertices, as well as their edge flow, make them difficult to modify after generation. As a future work, particle systems should be extended to be able to emit and propagate particles in 3D rather than in 2D. Moreover, this technique should be extended to represent others effects than those targeting FaVs, for example to visualize pollution, rain stains or corrosion effects on buildings or cars. More generally this technique could be adapted to modify the appearance of 3D objects based on time varying effects.

REFERENCES

[1] J. A. Bærentzen, M. K. Misztal, and K. Wełnicka. Converting Skeletal Structures to Quad-Dominant Meshes. *Computer & Graphics*, 36(5):555–561, 2012.

[2] V. Bevilacqua, G. Filograno, M. Fiorentino, and A. E. Uva. Early diagnosis of lung tumors by genetically optimized 3D-metaball malignancy metric. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion - GECCO Companion '12*, page 531, New York, New York, USA, 2012. ACM Press.

[3] J. F. Blinn. Simulation of Wrinkled Surfaces. *SIGGRAPH Comput. Graph.*, 12(3):286–292, Aug. 1978.

[4] J. F. Blinn. A Generalization of Algebraic Surface Drawing. *ACM Transactions on Graphics*, 1(3):235–256, jul 1982.

[5] J. Bloomenthal. Modeling the mighty maple. *ACM SIGGRAPH Computer Graphics*, 19(3):305–311, jul 1985.

[6] E. Bohl, O. Terraz, and D. Ghazanfarpour. Modeling fruits and their internal structure using parametric 3Gmap L-systems. *The Visual Computer*, 31(6-8):819–829, jun 2015.

[7] E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355, 1978.

[8] M. Cieslak, F. Boudon, S. Kenouche, M. Zanca, C. Goze-Bac, M. Génard, C. Godin, and N. Bertin. Generating 3D Volumetric Meshes of Internal and External Fruit Structure. *IV International Symposium on Models for Plant Growth, Environmental Control and Farm Management in Protected Cultivation - Hortimodel 2012*, 957:239–245, 2012.

[9] M. Cieslak, M. Génard, S. Kenouche, C. Goze-Bac, C. Godin, and N. Bertin. Towards a 3D virtual fruit model integrating fruit architecture and physiology. *Acta Horticulturae*, 1(1068):59–66, feb 2015.

[10] D. Fan, S. Liu, and Y. Wei. Fruit Ring Rot Simulation Based on Reaction-Diffusion Model. In *Proceedings of the 2013 International Conference on Virtual Reality and Visualization*, ICVRV '13, pages 199–205, Washington, DC, USA, 2013. IEEE Computer Society.

[11] Food And Agriculture Organization of the United Nation. Global food losses and food waste. http://www.fao.org/docrep/014/mb060e/mb060e.pdf. Accessed: 2015-10-10.

[12] D. R. Fowler, P. Prusinkiewicz, and J. Battjes. A collision-based model of spiral phyllotaxis. *ACM SIGGRAPH Computer Graphics*, 26(2):361–368, jul 1992.

[13] M. Fuhrer, H. W. Jensen, and P. Prusinkiewicz. Modeling hairy plants. *Graphical Models*, 68(4):333–342, jul 2006.

[14] M. Génard, N. Bertin, C. Borel, P. Bussieres, H. Gautier, R. Habib, M. Lechaudel, A. Lecomte, F. Lescourret, P. Lobit, and B. Quilot. Towards a virtual fruit focusing on quality: modelling features and potential uses. *Journal of Experimental Botany*, 58(5):917–928, jan 2007.

[15] M. Génard, N. Bertin, H. Gautier, F. Lescourret, and B. Quilot. Virtual profiling: a new way to analyse phenotypes. *The Plant Journal*, 62(2):344–355, jan 2010.

[16] J. Gu, C.-I. Tu, R. Ramamoorthi, P. Belhumeur, W. Matusik, and S. Nayar. Time-varying Surface Appearance: Acquisition, Modeling and Rendering. *ACM Trans. Graph.*, 25(3):762–771, July 2006.

[17] C. Y. Huang, W. T. Jheng, W. K. Tai, C. C. Chang, and D. L. Way. Procedural grape bunch modeling. *Computers and Graphics (Pergamon)*, 37(4):225–237, jun 2013.

[18] Z. Ji, L. Liu, and Y. Wang. B-Mesh: A modeling system for base meshes of 3D articulated shapes. *Computer Graphics Forum*, 29(7):2169–2177, 2010.

[19] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki. Harmonic coordinates for character articulation. *ACM Transactions on Graphics*, 26(3):71, jul 2007.

[20] J. T. Kider, S. Raja, and N. I. Badler. Fruit senescence and decay simulation. *Computer Graphics Forum*, 30(2):257–266, 2011.

[21] N. Loebnitz, G. Schuitema, and K. G. Grunert. Who Buys Oddly Shaped Food and Why? Impacts of Food Shape Abnormality and Organic Labeling on Purchase Intentions. *Psychology & Marketing*, 32(4):408–421, apr 2015.

[22] W. E. Lorensen and H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 163–169, New York, NY, USA, 1987. ACM.

[23] J. Lu, A. S. Georghiades, A. Glaser, H. Wu, L.-Y. Wei, B. Guo, J. Dorsey, and H. Rushmeier. Context-aware Textures. *ACM Trans. Graph.*, 26(1), Jan. 2007.

[24] National Fruit Collection. NFC Database. http://www.nationalfruitcollection.org.uk/index.php. Accessed: 2015-10-10.

[25] H. Nishimura, A. Hirai, T. Kawai, T. Kawata, I. Shirakawa, and K. Omura. Object Modeling by distribution function and a method of image generation. *Transactions of the Institute of Electronics and Communication Engineers of Japan*, J68- D(3):718–725, 1985.

[26] Organisation for Economic Co-operation and Development. International Standards for Fruit and Vegetables. http://www.oecd.org/tad/code/fruitandvegetables.htm. Accessed: 2015-10-10.

[27] H. Pan, H. Huo, G. Cui, and S. Chen. Modeling for Deformable Body and Motion Analysis: A Review. *Mathematical Problems in Engineering*, 2013:1–14, 2013.

[28] J. Pan, C. Zhao, X. Zhao, A. Hao, and H. Qin. Metaballs-based physical modeling and deformation of organs for virtual surgery. *The Visual Computer*, 31(6-8):947–957, may 2015.

[29] P. Prusinkiewicz, L. Mündermann, R. Karwowski, and B. Lane. The use of positional information in the modeling of plants. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*, pages 289–300, New York, New York, USA, 2001. ACM Press.

[30] Semeur.fr. Tomate coeur de boeuf. https://www.semeur.fr/wiki/index.php?title=Fichier:Tomate_coeur_de_boeuf_reif_red-2.jpg. Accessed: 2017-09-22.

[31] A. Verhulst, J.-M. Normand, C. Lombart, and G. Moreau. A study on the use of an immersive virtual reality store to investigate consumer perceptions and purchase behavior toward non-standard fruits and vegetables. In *2017 IEEE Virtual Reality, VR 2017, Los Angeles, CA, USA, March 18-22, 2017*, pages 55–63, 2017.

[32] S. A. Wainwright. *Axis and Circumference: The Cylindrical Shape of Plants and Animals*. Harvard University Press, 1988.

[33] W. E. Waterlander, Y. Jiang, I. H. M. Steenhuis, and C. N. Mhurchu. Using a 3d virtual supermarket to measure food purchase behavior: A validation study. *J Med Internet Res*, 17(4):e107, 2015.

[34] W. E. Waterlander, M. Scarpa, D. Lentz, and I. H. M. Steenhuis. The virtual supermarket: an innovative research tool to study consumer food purchasing behaviour. *BMC public health*, 11(1):589, 2011.